

03-24-00

EJ18764554305

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

March 23, 2000

Docket No. AUS000149US1

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of Inventor(s):

Maximino Aguilar
Sanjay Gupta
James Michael StaffordJC564 U.S. PTO
09/533301
03/23/00For: SYSTEM, APPARATUS AND METHOD FOR SUPPORTING MULTIPLE
PARTITIONS INCLUDING MULTIPLE SYSTEMS IN BOOT CODE

Enclosed are also:

- ☒ 21 Pages of Specification including an Abstract
☒ 9 Pages of Claims
☒ 8 Sheet(s) of Drawings
☒ A Declaration and Power of Attorney
☒ Form PTO 1595 and assignment of the invention to IBM Corporation

CLAIMS AS FILED

FOR	Number Filed		Number Extra		Rate		Basic Fee (\$690)
Total Claims	41	-20 =	21	X	\$ 18	=	\$378
Independent Claims	4	-3 =	1	X	\$ 78	=	\$78
Multiple Dependent Claims	0			X	\$260	=	\$
Total Filing Fee =							\$1146.00

- ☒ Please charge \$1146.00 to IBM Corporation, Deposit Account No. 09-0447.
☒ The Commissioner is hereby authorized to charge payment of the following fees associated with the communication or credit any over payment to IBM Corporation, Deposit Account No. 09-0447. A duplicate copy of this sheet is enclosed.
☒ Any additional filing fees required under 37CFR § 1.16.
☒ Any patent application processing fees under 37CFR § 1.17.

Respectfully,

Casimer K. Salys

Reg. No. 28,900

Intellectual Property Law Dept.

IBM Corporation

11400 Burnet Road 4054

Austin, Texas 75758

Telephone: (512) 823-0494

Docket No. AUS000149US1

**SYSTEM, APPARATUS AND METHOD FOR SUPPORTING
MULTIPLE PARTITIONS INCLUDING MULTIPLE OPERATING SYSTEMS
IN BOOT CODE**

5

BACKGROUND OF THE INVENTION

RELATED APPLICATION

The present application is related to commonly assigned
10 and co-pending U.S. Patent Application No. _____
(Attorney Docket No. AUS000150US1) entitled "System,
Apparatus and Method for Supporting Multiple File Systems
in Boot Code," which is hereby incorporated by reference.

15 **1. Technical Field:**

The present invention is directed to a system,
apparatus and method for supporting multiple file systems
in boot code. In particular, the present invention is
directed to a system, apparatus and method in which boot
20 code of a computer performs an identification of a file
system utilized by a boot disk and is capable of loading
an associated operating system based on the
identification of the file system.

25 **2. Description of Related Art:**

With known personal computers (PCs), an operating
system must be loaded by the system before file data can
be accessed by the personal computer. Modern computers
have programs stored in read only memory (ROM) which are
30 retained even when power to the computer is discontinued.
These programs, also known as **Basic Input Output Systems**
(BIOS), are used to load the operating system of the
personal computer. The BIOS is an essential set of

Docket No. AUS000149US1

routines in a PC which provides an interface between the operating system and the hardware.

On startup, the BIOS tests the system and prepares the computer for operation by querying its own small
 5 Complementary Metal Oxide Semiconductor (CMOS) memory bank for drive and other configuration settings. The BIOS searches for other BIOS's on plug-in boards and sets up pointers (interrupt vectors) in memory to access those other BIOS routines. It then loads a boot block and
 10 passes control to it.

The BIOS is capable of loading a single disk block into memory and passing control to it. This disk block is known as the boot block. A boot block is a reserved data block on a disk that is used to load the operating
 15 system. On startup, the BIOS loads the master boot record (MBR), which is in the boot block, into memory. The MBR contains pointers to the first sector, i.e. the boot sector, of the partition that contains the operating system. The boot sector contains the instructions that
 20 cause the computer to boot the operating system. The operating system is then utilized to access the computer's file system.

A file system is a collection of files, blocks, directories, and file descriptors located on one logical
 25 disk. A logical disk may be a physical disk, part of a physical disk, or several physical disks.

Figure 1 is a generalized illustration of a file system **100** for a hard disk. As shown in **Figure 1**, the file system **100** typically is comprised of a boot block
 30 **110**, a file system descriptor block **120**, a file descriptors block **130**, a file data blocks **140**. The boot block **110** has been described above.

The file system descriptor block **120** contains

Docket No. AUS000149US1

information identifying the total size of the file system, the size of the file descriptor block, the first free block, the location of the file descriptor of the root directory, the time the file system was created, last modified, and last used, and other system meta-data.

The file descriptor block **130** contains file descriptors which are data structures maintained by the operating system that contain all the meta-data the operating system needs to record about a file. The file

descriptors include such information as the last time the file was read, the last time the file was written, the length of the file, the location of the file data on the disk, the protection status of the file, and the like. The file data blocks **140** contain the actual file data.

As shown in **Figure 2**, in order to access file data on a computer system, the computer's BIOS, having a BIOS loader routine **210**, first loads the boot block **110**. The boot block **110**, i.e., the operating system loader **220**, contains the MBR which points to a location for the boot sector of the operating system. The operating system loader loads the boot sector and then passes control to the operating system **230**. The operating system **230** is then used to access the file data **240**.

Thus, known personal computers have programs stored in ROM, i.e. BIOS, which load an operating system based on a location of the operating system boot sector identified by the master boot record. Even if the personal computer contains a hard drive having multiple partitions with multiple operating systems, the master boot record determines which operating system is loaded by identifying the boot sector of the operating system to be loaded. The partition used during booting, and thus, the operating system that is booted, may be modified by

Docket No. AUS000149US1

changing the BIOS to indicate a different boot partition.

In view of the above, known personal computers are limited in that they may only load an operating system that is designated as the boot operating system in the master boot record. Thus, the operating system that is to be booted must be determined prior to a boot attempt.

The above description is applicable to personal computers. Network computers differ from personal computers in that network computers do not contain a BIOS. A network computer is a desktop computer that provides connectivity to intranets and/or the Internet. It is designed as a "thin client" that downloads all applications from the network server and obtains all of its data from and stores all changes back to the server.

The network computer (NC) is similar to a diskless workstation and does not have floppy or hard disk storage. The network computer may contain a minimal amount of memory in which a boot code and an operating system may be stored. The boot code is specific to the particular operating system and is similar to the operating system loader in the BIOS of a personal computer.

When a network operator wishes to add a network computer to an existing network, the operator must first configure the network computer by loading the appropriate boot code and operating system into the network computer for the particular file system used by the network. Thus, the network computer, once configured, is only able to operate with a network using a file system that is supported by the boot code and operating system loaded into the network computer.

If the network computer is to be used with a different network file system, the boot code and

operating system must be replaced with new boot code and a new operating system. Just as with the personal computer described above, the operating system that is to be loaded during a boot of the network computer must be determined prior to a boot attempt.

Thus, it would be advantageous to have a more versatile computer that is capable of determining which operating system to load at the time of a boot attempt.

SUMMARY OF THE INVENTION

5 The present invention provides a system, apparatus and method for supporting multiple file systems in boot code. The boot code according to the present invention searches a boot disk to determine the file systems used by the boot disk. There may be one or more file systems
10 on the boot disk. If there are multiple file systems, there are multiple partitions of the boot disk and may be multiple operating systems corresponding to the partitions.

Once the boot code determines the file systems used
15 on the boot disk, the boot code is able to identify corresponding operating systems for the file systems. Based on the identification of the operating systems, the boot code is able to locate a boot sector for the operating systems and load an appropriate operating
20 system. The boot code then turns over control of the computer system resources to the loaded operating system.

In the case where there are multiple partitions and multiple operating systems, the boot code according to the present invention is capable of selecting an
25 operating system from the multiple operating systems based on a selection criteria. In one embodiment, the selection criteria may be a priority assigned to the operating systems. In another embodiment, the selection may be based on a selection made by a user of the
30 computer system.

In the case where there are multiple operating systems using the same file system, the present invention is capable of determining which operating system to load by analyzing a boot block of the file system to determine

2025 RELEASE UNDER E.O. 14176

a corresponding operating system.

10

BRIEF DESCRIPTION OF THE DRAWINGS

5 The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed
10 description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 is an illustration of a known file system layout;

Figure 2 is a diagram illustrating the process for
15 loading an operating system for accessing file data;

Figure 3 is a diagram illustrating a distributed data processing system in which the present invention may be implemented;

Figure 4 is a diagram illustrating a network computer
20 in accordance with the present invention;

Figure 5 is a diagram illustrating a multiple file system architecture;

Figure 6 is a diagram illustrating partitions of a hard drive in which each partition contains a different
25 operating system;

Figure 7 is a flowchart outlining an exemplary operation of the present invention when selecting an operating system to load; and

Figure 8 is a flowchart outlining an exemplary
30 operation of the present invention when selecting an operating system to load from a multiple partition system.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

5

The present invention provides an apparatus and method in which an operating system that is to be loaded during booting of a computer is determined at the time of the boot attempt rather than prior to the boot attempt.

10 The present invention is applicable to both personal computers and network computers. However, for descriptive purposes, only a network computer embodiment will be described herein. Those of ordinary skill in the art will readily recognize the application of the
15 principles of the present invention to personal computers.

With reference now to the figures, and in particular with reference to **Figure 3**, a pictorial representation of a distributed data processing system is depicted in which
20 the present invention may be implemented. Distributed data processing system **300** is a network of computers in which the present invention may be implemented. Distributed data processing system **300** contains network
310, which is the medium used to provide communications
25 links between various devices and computers connected within distributed data processing system **300**. Network **310** may include permanent connections, such as wire or fiber optic cables, or temporary connections made through telephone connections.

30 In the depicted example, server **350** is connected to network **310**, along with storage unit **360**. In addition, network computers **320**, **330** and **340** are also connected to network **310**. The storage unit **360** and printers **370-380**

DOCKET FOR FILE

Docket No. AUS000149US1

are depicted to represent network resources that may be accessed by the network computers **320-340** via the server **350**.

For purposes of this application, a network computer
5 is any diskless workstation coupled to a network which
downloads all applications from a network server and
obtains all of its data from and stores all data changes
back to the network server **350**. A network computer may
have some capability for the addition of expansion boards
10 such as sound cards, graphics adapters, input/output
boards, and the like. This capability may be limited to
reduce the cost and complexity of the network computer.

In the depicted example, server **350** provides data,
such as boot files, operating system images and
15 applications, to network computers **320-340**. Network
computers **320-340** are clients to server **350**. Distributed
data processing system **300** may include additional
servers, clients, and other devices not shown.
Distributed data processing system **300** also includes
20 printer **370**. Client devices, such as network computers
320-340, may print to printer **370**, which is attached to
server **350**, or to printer **380**, which is a network printer
that does not require connection to a computer for
printing documents. In the depicted example, distributed
25 data processing system **300** is the Internet, with network
310 representing a worldwide collection of networks and
gateways that use the TCP/IP suite of protocols to
communicate with one another. At the heart of the
Internet is a backbone of high-speed data communication
30 lines between major nodes or host computers consisting of
thousands of commercial, government, education, and other
computer systems that route data and messages. Of

Docket No. AUS000149US1

course, distributed data processing system **300** also may be implemented as a number of different types of networks such as, for example, an intranet or a local area network. **Figure 3** is intended as an example and not as
5 an architectural limitation for the processes of the present invention.

The network computers **320-340** contain boot code that permits the network computers **320-340** to operate under a plurality of operating systems. Thus, the network
10 computers **320-340** are generic network computers. The network computers **320-340**, when booting, search the file system descriptor block of the server **350** to determine the type of file system being used by the server **350**. Such determination may be based on file system
15 identifiers that are located in the file system descriptor block from which the file system may be discerned.

Once the file system is identified, the boot code contains information for identifying an operating system associated with the file system. Based on the
20 identification of the operating system, the boot code identifies a boot block associated with the operating system. This determination may be made by consulting, for example, correspondence information included as part
25 of the boot code. Once the operating system and boot block are determined, the boot code loads the boot block and turns over control to the loaded operating system. Thereafter, the network computer is capable of accessing, modifying, and storing file data in the file system of
30 the server **350**.

With reference now to **Figure 4**, a block diagram of a data processing system in which the present invention may be implemented is illustrated. Data processing system

Docket No. AUS000149US1

400 is an example of a network computer, such as network computer **320**. Data processing system **400** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a
5 PCI bus, other bus architectures, such as Micro Channel and ISA, may be used. Furthermore, although a single processor data processing system is shown in **Figure 4**, the invention is applicable to multiple processor systems as well as single processor systems.

10 Processor **402** and main memory **404** are connected to PCI local bus **406** through PCI bridge **408**. PCI bridge **408** may also include an integrated memory controller and cache memory for processor **402**. Additional connections to PCI local bus **406** may be made through direct component
15 interconnection or through add-in boards. As mentioned above, the ability to provide additional connections may be limited to reduce cost and complexity of the network computer.

In the depicted example, local area network (LAN)
20 adapter **410** and expansion bus interface **414** are connected to PCI local bus **406** by direct component connection. In contrast, audio adapter **416**, graphics adapter **418**, and audio/video adapter (A/V) **419** are connected to PCI local bus **406** by add-in boards inserted into expansion slots.
25 Expansion bus interface **414** provides a connection for a keyboard and mouse adapter **420** and additional memory **424**.

An operating system runs on processor **402** and is used to coordinate and provide control of various components within data processing system **400**. The
30 operating system may be a commercially available operating system, such as OS/2, which is available from International Business Machines Corporation. "OS/2" is a

Docket No. AUS000149US1

trademark of International Business Machines Corporation.

The particular operating system that runs on the processor **402** is determined by boot code, as will be described in greater detail hereafter. Instructions for the operating system and applications or programs are obtained from network servers via the network interface **410**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 4** may vary depending on the implementation. The depicted example is not meant to imply architectural limitations with respect to the present invention.

With the present invention, non-volatile memory **460** contains boot code capable of booting a plurality of operating systems. However, the boot code according to this invention may be stored in other non-volatile memory devices, such as memory **424** or the main memory **404** without departing from the spirit and scope of the present invention. The boot code according to the present invention first identifies the file system being used by the network server with which the network computer **400** is communicating. This identification may be performed, for example, by reading a file system identifier in the file system descriptor block **120**. The network computer **400** reads the file system descriptor block **120** from the network server via the network interface **410** and identifies the file system being used by the network server based on the identifier in the file system descriptor block **120**.

The file system descriptor block **120** is also referred to as a file system super block. The file system super block contains information about the file system. In order to determine the file system being

Docket No. AUS000149US1

used, the super block is loaded by the boot code and the file system is identified by the information, such as an identifier, contained in the super block.

Based on the identification of the file system being
5 used by the network server, the boot code identifies an
operating system kernel location in a boot block of the
memory **424**, for example, for the operating system
associated with the file system. Most operating systems
used for Network computers have a preferred file system.
10 For example FFS is the preferred file system for NetNSD,
ext2FS is the preferred file system for Linux etc. Thus
based on the filesystem it can be determined which
operating system to load. Based on the identification of
the operating system, a boot block corresponding to the
15 operating system is identified, for example, based on
information stored in the boot code. The network
computer **400** then loads operating system instructions
into main memory **404** and turns control over to the loaded
operating system.

20 Because the boot code of the present invention first
identifies the file system of the network server with
which the network computer is communicating, and
identifies a corresponding operating system, the need for
multiple boot codes is eliminated. Thus, with the
25 present invention, a network computer need not be
pre-configured to be used with a particular operating
system. Rather, the present invention provides a generic
network computer having the boot code, according to the
present invention, stored in memory thereon. The boot
30 code allows the network computer to be used with any
operating system so long as the boot code is capable of
identifying the file system and a corresponding operating
system used by the network servers. Thus, the boot code

Docket No. AUS000149US1

according to the present invention provides the capability of determining the operating system to be loaded at the time of the boot attempt and does not require that the operating system be determine a priori.

5 Furthermore, the present invention provides a network computer that may load different operating systems based on the file system of the particular network server with which the network computer is communicating. Thus, if the network computer of the
10 present invention is communicating with a first network server having a FAT32 file system, the network computer will boot the Windows 98™ operating system. If the network computer then communicates with a second network server having a High Performance File System (HPFS), the
15 network computer will boot the OS/2™ operating system. Thus, the present invention provides a network computer capable of adapting its loaded operating system to the particular file system used by the network server with which it is presently communicating.

20 In some cases, the same file system may be utilized by a plurality of operating systems. For example, FAT is a file system that may be used by DOS, Windows and the OS/2 operating systems. The present invention is capable of determining which of the three operating systems to
25 load.

 The boot code of the present invention first identifies the file system in the manner set forth above. Thereafter, the boot code determines that there may be multiple operating systems that use the identified file
30 system in a manner similar to the identification of the operating system described above. The boot code then looks at the boot block of the file system on the network server to determine which operating system corresponds to

Docket No. AUS000149US1

the file system of the network server.

In other cases, a network server may contain multiple operating systems located in multiple partitions of the network server hard disk. **Figure 5** illustrates a hard disk having three partitions **510**, **520** and **530**. Each partition **510-530** contains a different file system, and thus, a different operating system. Partition **510** contains a FAT32 file system and a Windows 98™ operating system. Partition **520** contains an NT File System (NTFS) and a Windows NT™ operating system. Partition **530** contains a High Performance File System (HPFS) and an OS/2™ operating system.

In the case of multiple partitions, as shown in **Figure 6**, the present invention provides a boot code that identifies the file systems **640-660** in each of the partitions **610-620**. The partitions **610-620** themselves are identified based on information, such as, for example, a partition table and start and end logical sector information for each partition, stored in sector zero of the hard disk, in a manner generally known in the art. The file systems **640-660** are identified in the manner set forth above by reading file system descriptor blocks, i.e. super blocks, of each of the partitions **610-630**. Once the file systems **640-660** are identified, the corresponding operating systems may be identified in a manner similar to that described above with regard to **Figures 3 and 4**.

Since only one operating system may be loaded at one time without the use of emulation, a determination must be made as to which operating system from which partition should be loaded. This determination may be made, for example, based on a prioritization scheme wherein the priorities of the various types of operating systems are

Docket No. AUS000149US1

stored in the boot code of the network computer. Thus, if the file system identification of each of the partitions indicates that three operating systems Windows 98™, Windows NT™ and OS/2™ are present, a prioritization
5 scheme may indicate that Windows NT™ should be loaded. In the case of an error during loading of the operating system, the boot code may provide for loading of the next highest priority operating system.

Once it is determined which operating system is to
10 be loaded, the boot code identifies the location of the operating system kernel, i.e. the boot sector for the operating system. The boot sector of the operating system is typically the first sector of the hard disk partition corresponding to the file system. The boot
15 code then loads the operating system kernel and turns over control to the operating system. Thereafter, the file data of the corresponding operating system may be accessed, modified and stored.

The priorities of the operating systems may be
20 modified by a user of the network computer such that a different operating system from a currently highest priority operating system is provided with the highest priority. The modifications to the priorities of the operating systems may be performed, for example, based on
25 a menu provided in the boot code. The menu may be displayed to the user, in accordance with instructions stored in the boot code, to identify the possible operating systems that may be loaded. The user may then select an operating system from the displayed operating
30 systems using an input device, such as a keyboard, mouse, or the like.

Thus, the present invention is capable of supporting multiple partitions having multiple operating systems.

Docket No. AUS000149US1

In addition, the present invention is capable of selecting from a plurality of possible operating systems based on a priority scheme.

Figure 7 is a flowchart outlining an exemplary operation of the present invention when selecting an operating system to load. As shown in **Figure 7**, the operation starts with a search of the boot disk to determine the file system being used (step **710**). The boot disk may be a hard disk in a personal computer, a hard disk on a server, a network computer memory, or the like, from which the operating system is to be booted.

The file system is then identified and the location of the operating system is identified based on the identification of the file system (step **720**). The operating system kernel at the identified operating system location is loaded (step **730**) and control is then turned over to the operating system (step **740**). The operation then ends.

Figure 8 is a flowchart outlining the operation of the present invention when selecting an operating system to load from a multiple partition system. As shown in **Figure 8**, the operation starts with a search of each of the partitions of the system for their respective file systems (step **810**). The found file systems are then identified (step **820**) and the operating systems associated with the found file systems are determined (step **830**).

The priority of the operating systems is then determined (step **840**) and the location of the operating system kernel for the highest priority operating system is identified (step **850**). The kernel for the highest priority operating system is loaded (step **860**) and control is turned over to the operating system (step **870**). The

Docket No. AUS000149US1

operation then ends.

Thus, as described above, the present invention provides a versatile computer system having a boot code capable of supporting various operating systems. The
5 operating system that is loaded during a boot operation is determined at the time of the boot attempt rather than prior to the boot attempt. The determination is based primarily on the file system used by the boot disk. The boot disk may be located on the computer system having the
10 boot code or may be on a remote computer, such as a network server.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary
15 skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of
20 signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, CD-ROMs, and transmission-type media such as digital and analog communications links.

25 The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in
30 the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for

various embodiments with various modifications as are suited to the particular use contemplated.

Docket No. AUS000149US1

CLAIMS:

What is claimed is:

- 5 1. A method, in a data processing system, for loading an operating system, comprising:
- searching, by a boot code, a set of partitions of a storage device for file systems resulting in found file systems;
- 10 identifying, by the boot code, a plurality of operating systems associated with the found file systems, each one of the file systems being associated with one or more of the plurality of operating systems;
- selecting, by the boot code, one operating system of
- 15 the plurality of operating systems; and
- loading, by the boot code, operating system instructions of the selected one operating system.
2. The method of claim 1, further comprising:
- 20 identifying each of the found file systems, wherein each of the found file systems are identified by loading a file system descriptor block associated with each of the found file systems.
- 25 3. The method of claim 1, wherein identifying a plurality of operating systems associated with the found file systems includes identifying a preferred operating system for each found file system based on file system information in a file system descriptor block for each of
- 30 the found file systems.
4. The method of claim 1, wherein identifying a plurality of operating systems includes identifying a boot sector on the storage device associated with an

Docket No. AUS000149US1

identified operating system for each of the plurality of operating systems.

5 5. The method of claim 1, wherein loading operating system instructions includes loading an operating system kernel in a boot sector of the storage device.

10 6. The method of claim 1, wherein the method is implemented in a network computer.

15 7. The method of claim 1, wherein identifying a plurality of operating systems associated with the found file systems includes identifying a second plurality of operating systems corresponding to one file system of the found file systems and selecting one of the second plurality of operating systems as an operating system associated with the one file system.

20 8. The method of claim 7, wherein selecting one of the second plurality of operating systems includes looking at a boot block of the one file system and selecting an operating system from the second plurality of operating systems based on information in the boot block of the one file system.

25 9. The method of claim 1, wherein the method is implemented by a network computer and wherein the plurality of file systems are located on a network server.

30 10. The method of claim 1, wherein the method is implemented in a personal computer and wherein the storage device is one of a hard disk, a floppy disk, and

Docket No. AUS000149US1

an optical disk accessed by the personal computer.

11. The method of claim 1, wherein selecting one
operating system of the plurality of operating systems
5 includes determining priorities of the plurality of
operating systems and selecting an operating system
having a highest priority.

12. The method of claim 11, wherein the priorities of
10 the plurality of operating systems are modifiable by a
user.

13. The method of claim 11, wherein the priorities of
the plurality of operating systems are modifiable by a
15 user via a menu.

14. A data processing apparatus for loading an operating
system, comprising:

20 a processor unit; and
a first storage device coupled to the processor
unit, the first storage device storing boot code that is
executed by the processor unit to perform the functions:

identifying a plurality of file systems on a second
storage device, each one of the plurality of file systems
25 being associated with a partition of the second storage
device;

identifying a plurality of operating systems
associated with the plurality of file systems, each one
of the plurality of file systems being associated with
30 one or more of the plurality of operating systems;

selecting one operating system of the plurality of
operating systems; and

loading operating system instructions of the

Docket No. AUS000149US1

selected one operating system.

15. The apparatus of claim 14, wherein the plurality of
file systems are identified by loading a file system
5 descriptor block associated with each one of the file
systems.

16. The apparatus of claim 14, wherein identifying a
plurality of operating systems associated with the
10 plurality of file systems includes identifying a
preferred operating systems based on file system
information in a file system descriptor block for each
one of the plurality of file systems.

17. The apparatus of claim 14, wherein identifying a
plurality of operating systems includes identifying a
boot sector on the storage device associated with an
identified operating system for each of the plurality of
operating systems.

18. The apparatus of claim 14, wherein loading operating
system instructions includes loading an operating system
kernel in a boot sector of the storage device.

19. The apparatus of claim 14, wherein the data
processing apparatus is a network computer.

20. The apparatus of claim 14, wherein identifying a
plurality of operating systems associated with the
30 plurality of file systems includes identifying a second
plurality of operating systems corresponding to one file
system of the plurality of file systems and selecting one
of the second plurality of operating systems as an

Docket No. AUS000149US1

operating system associated with the one file system.

21. The apparatus of claim 20, wherein selecting one of the second plurality of operating systems includes
5 looking at a boot block of the one file system and selecting an operating system from the second plurality of operating systems based on information in the boot block of the one file system.

10 22. The apparatus of claim 14, wherein the data processing apparatus is a network computer and wherein the plurality of file systems are located on a network server.

15 23. The apparatus of claim 14, wherein the data processing apparatus is implemented in a personal computer and wherein the second storage device is one of a hard disk, a floppy disk, and an optical disk accessed by the personal computer.

20 24. The apparatus of claim 14, wherein selecting one operating system of the plurality of operating systems includes determining priorities of the plurality of operating systems and selecting an operating system
25 having a highest priority.

25. The apparatus of claim 24, wherein the priorities of the plurality of operating systems are modifiable by a user.

30 26. The apparatus of claim 24, wherein the priorities of the plurality of operating systems are modifiable by a user via a menu.

Docket No. AUS000149US1

27. The apparatus of claim 14, wherein the processor unit includes at least one processor.

5 28. A computer program product, in a computer readable medium, for loading an operating system, comprising:

first instructions for identifying a plurality of file systems on a storage device, each one of the plurality of file systems being associated with a

10 partition of the storage device;

second instructions for identifying a plurality of operating systems associated with the plurality of file systems, each one of the plurality of file systems being associated with one or more of the plurality of operating
15 systems;

third instructions for selecting one operating system of the plurality of operating systems; and

fourth instructions for loading operating system instructions of the selected one operating system.

20

29. The computer program product of claim 28, wherein the first instructions include instructions for loading a file system descriptor block associated with each one of the file systems.

25

30. The computer program product of claim 28, wherein the second instructions include instructions for identifying a preferred operating systems based on file system information in a file system descriptor block for
30 each one of the plurality of file systems.

31. The computer program product of claim 28, wherein the second instructions include instructions for

DOCKET FOR FILE

Docket No. AUS000149US1

identifying a boot sector on the storage device associated with an identified operating system for each of the plurality of operating systems.

5 32. The computer program product of claim 28, wherein the fourth instructions include instructions for loading an operating system kernel in a boot sector of the storage device.

10 33. The computer program product of claim 28, wherein the second instructions include instructions for identifying a second plurality of operating systems corresponding to one file system of the plurality of file systems and selecting one of the second plurality of
15 operating systems as an operating system associated with the one file system.

20 34. The computer program product of claim 33, wherein the instructions for selecting one of the second plurality of operating systems include instructions for looking at a boot block of the one file system and selecting an operating system from the second plurality of operating systems based on information in the boot block of the one file system.

25 35. The computer program product of claim 28, wherein the third instructions include instructions for determining priorities of the plurality of operating systems and selecting an operating system having a
30 highest priority.

36. The computer program product of claim 35, wherein the priorities of the plurality of operating systems are

Docket No. AUS000149US1
modifiable by a user.

37. The computer program product of claim 35, wherein
the priorities of the plurality of operating systems are
5 modifiable by a user via a menu.

38. A data processing system for loading an operating
system, comprising:

first identifying means for identifying a plurality
10 of file systems on a storage device, each one of the
plurality of file systems being associated with a
partition of the storage device;

second identifying means for identifying a plurality
of operating systems associated with the plurality of
15 file systems, each one of the plurality of file systems
being associated with one or more of the plurality of
operating systems;

selecting means for selecting one operating system
of the plurality of operating systems; and

20 loading means for loading operating system
instructions of the selected one operating system.

39. The data processing system of claim 38, wherein the
25 selecting means includes priority means for determining
priorities of the plurality of operating systems and
selecting an operating system having a highest priority.

40. The data processing system of claim 39, wherein the
30 priorities of the plurality of operating systems are
modifiable by a user.

41. The data processing system of claim 39, wherein the

Docket No. AUS000149US1

priorities of the plurality of operating systems are modifiable by a user via a menu.

2025 RELEASE UNDER E.O. 14176

Docket No. AUS000149US1

ABSTRACT OF THE DISCLOSURE

SYSTEM, APPARATUS AND METHOD FOR SUPPORTING
5 MULTIPLE PARTITIONS INCLUDING MULTIPLE OPERATING SYSTEMS
IN BOOT CODE

A system, apparatus and method for supporting multiple file systems in boot code of a computer. The boot code according to the present invention first identifies file systems used by a boot disk and then identifies operating systems associated with the identified file systems. Based on the identified operating systems, the boot sector for an appropriate operating system is located and loaded. Thereafter, the boot code relinquishes control to the loaded operating system. The boot code is capable of supporting multiple file systems, multiple operating systems located in a plurality of partitions of a boot disk, and multiple operating systems using the same file system.

The diagram illustrates a file system structure as a horizontal bar divided into four sections. From left to right, the sections are labeled: "Boot Block", "File System Descriptor", "File Descriptors", and "File Data Blocks". Below each section, a curly bracket points to a numerical label: "110" for the Boot Block, "120" for the File System Descriptor, "130" for the File Descriptors, and "140" for the File Data Blocks.

AUS000149US1
(Sheet 1 of 8)

PRIOR ART

AUS000149US1
(Sheet 2 of 8)

PRIOR ART

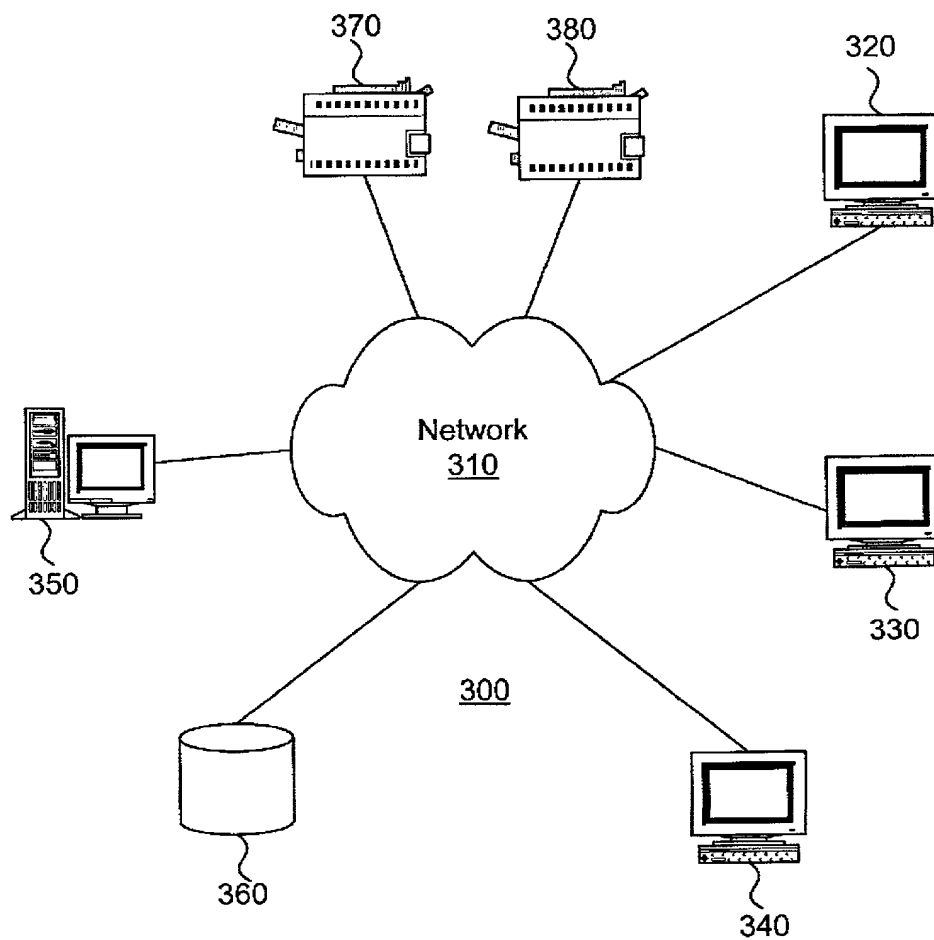


Figure 3

AUS000149US1
(Sheet 3 of 8)

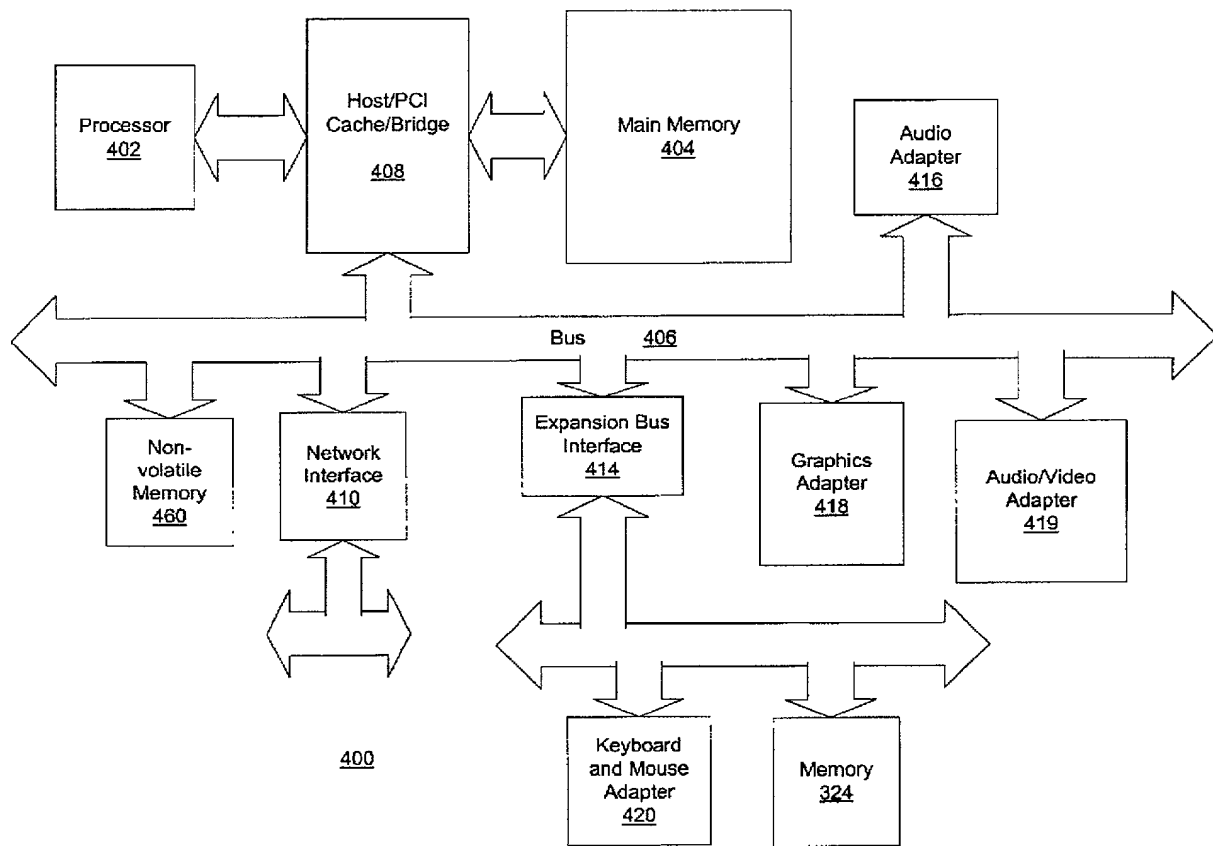


Figure 4

AUS000149US1
(Sheet 4 of 8)

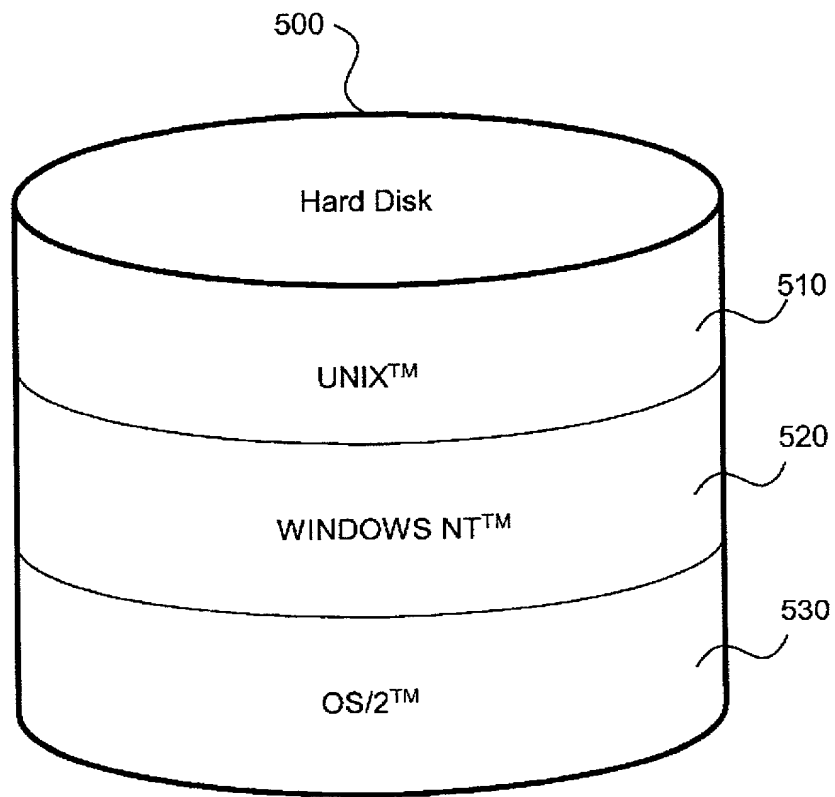


Figure 5

AUS000149US1
(Sheet 5 of 8)

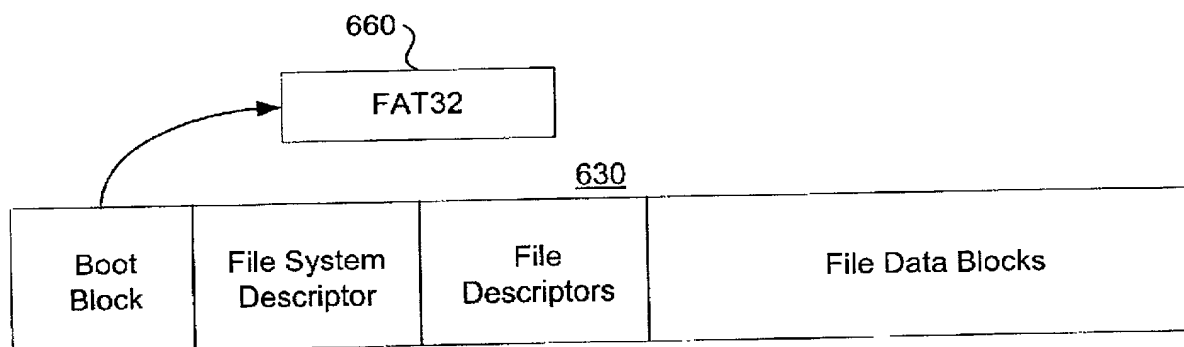
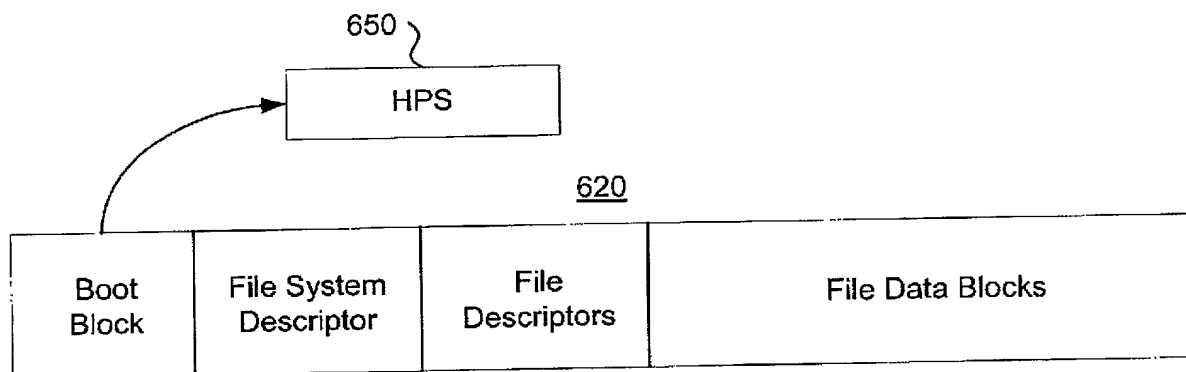
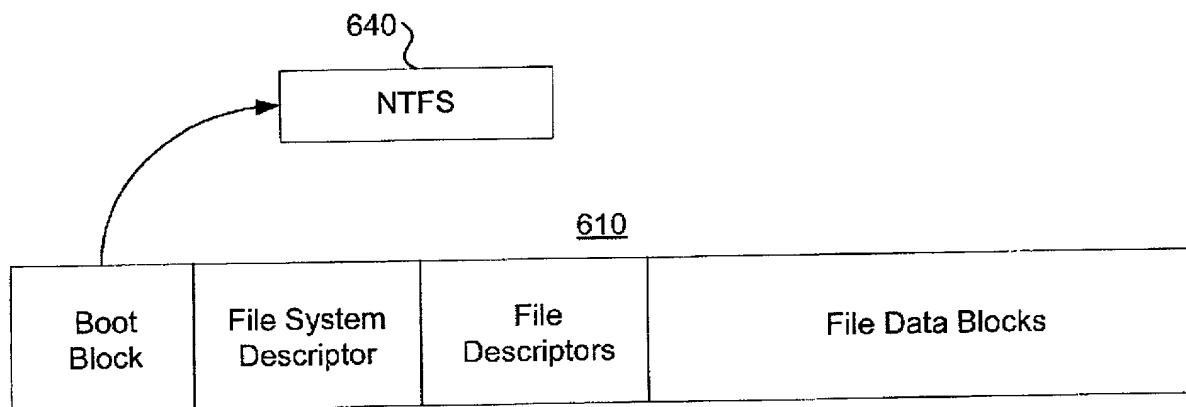


Figure 6

Figure 7

AUS000149US1
(Sheet 7 of 8)

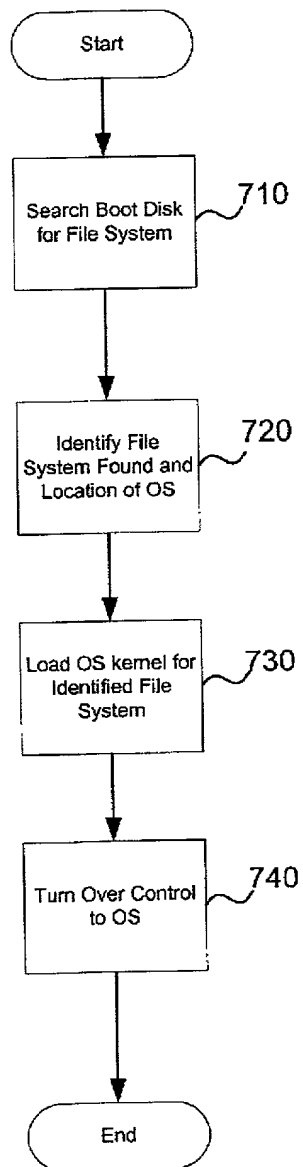
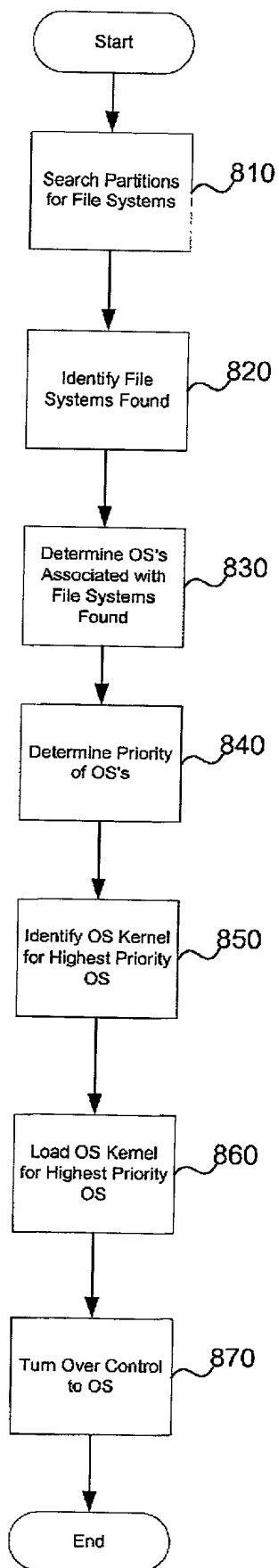


Figure 8

AUS000149US1
(Sheet 8 of 8)



**DECLARATION AND POWER OF ATTORNEY FOR
PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

SYSTEM, APPARATUS AND METHOD FOR SUPPORTING MULTIPLE PARTITIONS INCLUDING MULTIPLE OPERATING SYSTEMS IN BOOT CODE

the specification of which (check one)

X is attached hereto.

— was filed on _____
as Application Serial No. _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Priority Claimed

____ Yes ____ No
(Number) (Country) (Day/Month/Year)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial #) (Filing Date) (Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; Thomas E. Tyson, Reg. No. 28,543; James H. Barksdale, Jr., Reg. No. 24,091; Casimer K. Salys, Reg. No. 28,900; Robert M. Carwell, Reg. No. 28,499; Douglas H. Lefevre, Reg. No. 26,193; Jeffrey S. LaBaw, Reg. No. 31,633; David A. Mims, Jr., Reg. 32,708; Volel Emile, Reg. No. 39,969; Anthony V. England, Reg. No. 35,129; Leslie A. Van Leeuwen, Reg. No. 42,196; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Marilyn S. Dawkins, Reg. No. 31,140; Mark E. McBurney, Reg. No. 33,114; Duke W. Yee, Reg. No. 34,285; Colin P. Cahoon, Reg. No. 38,836; Rudolph J. Buchel, Reg. No. 43,448; and Stephen R. Loe, Reg. No. 43,757; Stephen J. Walder, Reg. No. 41,534; Charles D. Stepps, Jr., Reg. No. 45,880.

Send correspondence to: Duke W. Yee, Carstens, Yee & Cahoon, LLP, P.O. Box 802334, Dallas, Texas 75380 and direct all telephone calls to Duke W. Yee, (972) 367-2001

FULL NAME OF SOLE OR FIRST INVENTOR: Maximino Aguilar

INVENTORS SIGNATURE: Maximino Aguilar DATE: March 21, 00

RESIDENCE: 2301 S. MoPac #323
Austin, Texas 78746

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE

FULL NAME OF SECOND INVENTOR: Sanjay Gupta

INVENTORS SIGNATURE: Sanjay Gupta DATE: March 23, 2000

RESIDENCE: 11900 Hobby Horse Ct. #312
Austin, Texas 78758

CITIZENSHIP: India

POST OFFICE ADDRESS: SAME AS ABOVE

DOCKET NUMBER: AUS000149US1

FULL NAME OF THIRD INVENTOR: James Michael Stafford

INVENTORS SIGNATURE: James Michael Ruff DATE: March 21, 2000

RESIDENCE: 1060 Red Bud Lane #1060
Round Rock, Texas 78664

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE

Page 3 of 3